***Project 2: Minigame***

## Abstract:

The Minigame Project is an implementation of a game circuit. One LED on the Summer STEM LED matrix is controllable by stepping it from left to right. The LED constantly “moves” down. Separately, there is a row of LEDs displaying a randomly generated pattern. As the LED moves down, the object of the game is to AVOID the target LEDs that are on.

If the moving LED “hits” a target LED that is on, the “Game Over” light turns on and movement stops until the game is reset with a button.

On Teams, you will find a video description of this project. This project should be implemented in Logisim, approved by an instructor, and then built physically. Depending on your time constraints, you may choose which “packages” of this project you will complete.

**“Base Package”:** Implement the game as described above and shown in the video, minus the collision detection that enables the game to detect when you’ve lost and stop.

**“Upgraded Package”:** The base package, plus the collision detection and game stopping. (recommended!)

**“Deluxe Package”:** The upgraded package, but with a 5x5 playable area instead of 4x5 (columns x rows). *Note that if you choose this package, your initial design will likely be slightly different, so you can’t easily start with the base and modify it to deluxe.*

A great option is to implement a more complex design in Logisim but only physically build the simpler one.

Allowed Parts: Anything in your kit!

**Required**: pushbuttons (x2), LED Matrix, two breadboards, battery pack.

**Recommended**: CD4029, CD4013 (x2), various combinational logic gates.

## Design Specifications:

## ***Your design must be approved by an instructor.***

## Design Process:

1. Draw an I/O diagram. We’ll go over this in class.

1 ->

5 rows ->

4 columns ->

LED MATRIX 4x5

Target Row (4LEDs)

4 ->

Game Over (1 LED)

RESET ->

STEP ->

1. Draw a functional block diagram. Show this to an instructor – your specific block diagram will depend on which version you implement (base/upgraded/deluxe).
2. Systematically design each block, using the information we have taught so far. For example: A truth table and K-maps are helpful for collision detection. Another example: The STEP input controls the horizontal position of your LED in the matrix. What circuit have you learned to “shift” a bit? Have an instructor review your design.
3. Simulate as much of your design as you can in Logisim. The exact components are not available (LED Matrix, CD4029, etc) but you can approximate their functions in Logisim.
4. If you want and have time, simulate in TinkerCAD. Again, the exact components aren’t all available.

## Project Tips:

* Good work starts with showing your work!
  + Write down your Karnaugh maps and Boolean expressions on paper.
  + Using an Excel sheet to keep track of your work may be useful.
  + Diagram your project EXACTLY as you will build it. The CD4029 is NOT the same as the counters in Logisim or TinkerCAD. Pay attention to the specific control pins!
* Work smarter, not harder.
  + Use circuit design software like *Logisim* or *TinkerCAD Circuits* to plan or test out your system. May also be useful for debugging broken circuits.
  + Determine early on whether you are designing a light-on-1 or light-on-0 system.
  + There are online calculators for Boolean logic – take advantage of them.
* Don’t be afraid to ask for help!
  + Consult your classmates and instructors if you get stuck.
* Follow good practices while breadboarding.
  + Be consistent with wire colors
  + Keep wires running across only either rows or columns (never diagonally or curved)
  + Good practices make it easy to debug your circuit.
  + Spacing out components properly helps – don’t make spaces between chips too wide or to narrow – 4 to 5 pins should be enough. For DIP switches, space 8 pins.
  + Don’t put more than one wire into one breadboard hole.
* Optimize your system! Reuse outputs and/or transform them.
* Don’t overpower your components!
  + CMOS chips and switches can be run from your 4xAA battery pack.
  + The LED Matrix has built-in current limiting resistors, which should be sufficient. If your chips are getting too warm, add 100 resistors to each row or column of the Matrix.
* You may need to debounce the pushbutton switches. Some are more “bouncy” than others…See slides from Wednesday of Week 2.

**Column Control**

2x CD4013 D-ff

**Row Control**

1 CD4029 Counter

1 CD4051 Mux/Demux

1 AND Gate

1 OR Gate

**Collision Detection**

4 AND Gates (CD4081)

3 OR Gates (CD4071)

**Target Row**

1 CD4029 Counter

1 AND

**Clocks**

2 555 Timers

**TOTAL**

**10 CHIPS**

* **2 CD4081**
* **1 CD4071**
* **1 CD4051 MUX/DEMUX**
* **2 CD4029 COUNTERS**
* **2 CD4013 D-FF**
* **2 555 TIMERS**